UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de la Telecomunicación

TRABAJO FIN DE GRADO

Estudio de detección de ciberataques en Internet mediante algoritmos de clasificación de parámetros de tráfico

Javier Sánchez Caballero

Tutor: Luis de Pedro Sánchez

Ponente Jorge Enrique López de Vergara

Junio 2019

Estudio de detección de ciberataques en Internet mediante algoritmos de clasificación de parámetros de tráfico

AUTOR: Javier Sánchez Caballero
TUTOR: Luis de Pedro Sánchez

Dpto. Tecnología Electrónica y de Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2019

Resumen

Cada día son más los usuarios que utilizan Internet, lo cual se puede traducir en un mayor número de víctimas para los ciberdelincuentes. Debido a esto surge la necesidad de aumentar nuestra protección ante los infinitos tipos de ataques que podemos sufrir. Estos ataques son cada vez más avanzados por lo que también deben serlo los algoritmos que utilicemos. El primer paso para conseguir dicho objetivo es mejorar el porcentaje de detección de dichos ataques, ya que la mayoría de ellos pasan desapercibidos. Muchos de estos se producen por interacción del usuario, por lo que es de vital importancia que todo el mundo conozca los peligros que existen en la red.

El estudio realizado en este trabajo tiene como tarea principal ayudarnos a determinar los parámetros de tráfico que mayor porcentaje de aciertos van a proporcionarnos a la hora de clasificar los flujos por medio de árboles de decisión. Se van a utilizar dichos algoritmos, ya que permiten identificar de una manera más visual las decisiones y los caminos que toman cada uno de los flujos de datos en función del parámetro escogido. De entre los numerosos árboles de decisión que existen hemos escogido el árbol J48 para realizar el estudio debido a su buen desempeño a la hora de detectar intrusiones en un sistema.

El conjunto de datos de entrada, proporcionado por la universidad de Granada, está formado por tráfico real extraído de una red junto a diferentes tipos de ataques generados sintéticamente. El análisis consistirá en determinar de manera justificada la elección de los parámetros de tráfico e intentar maximizar el porcentaje de aciertos del algoritmo mediante diferentes técnicas.

Palabras clave

J48, Weka, ciberataques, árboles de decisión, tráfico de Internet

Abstract

Every day more and more users are connected to the Internet, which would translate into an increasing number of potential victims for cybercriminals. Therefore, there is an urgent need to increase our protection against the infinite types of attacks that we can suffer. These attacks are increasingly advanced, so should the algorithms we use. The first step to achieve this goal is to improve the percentage of detection of such attacks, since most of them go unnoticed. Many of those attacks are produced by user interaction, so it is vital that everyone knows the dangers that exist in the network.

The study carried out in this work has as its main task to help us determine the traffic parameters with the greatest success rate when classifying flows through decision trees. Among the numerous decision trees that exist, we have chosen the J48 tree to perform the study due to its good performance when detecting intrusions in a system.

The input data set, provided by the University of Granada, consists of real traffic extracted from a network together with different types of synthetically generated attacks. The analysis will consist of determining the election of the traffic parameters and trying to maximize the percentage of correct answers of the algorithm by means of different techniques.

Keywords

J48, Weka, cyberattacks, decision trees, Internet’s traffic

Agradecimientos

Agradezco a todas las personas que me han apoyado y me han ayudado a conseguir mis objetivos. Gracias a mi familia y a mis amigos por darme apoyo moral siempre. Gracias a mis compañeros de clase, y en especial a mi novia, por haber formado tan buen equipo y haber sacado lo mejor de esta etapa.

Por último, agradecer la ayuda a mis tutores, sin los cuales no habría sido posible concluir este trabajo.

INDICE DE CONTENIDOS

INDICE DE ILUSTRACIONES

INDICE DE TABLAS

GLOSARIO

DoS	Denegación de Servicio
DDoS	Denegación de Servicio Distribuido
ISP	Proveedor de Servicios de Internet

1 Introducción

En este capítulo se introducirán las razones por las que se realiza este TFG, y se definirá el plan de acción que se llevará a cabo para cumplir los objetivos propuestos y obtener las conclusiones que validarán la realización del estudio. Esta sección se divide en:

- Motivación
- Objetivos
- Fases de realización
- Organización de la memoria

1.1 Motivación

Cada año aumentan más los casos de ciberataques dirigidos a personas, e incluso a compañías enteras, las cuales al no tener la preparación suficiente han perdido millones de euros debido a esto. Diariamente se descubren nuevos tipos de ataques y los ciberdelincuentes están cada vez mejor formados, por lo que necesitamos encontrar una solución para poder estar suficientemente protegidos ante dichos peligros.

Para poder solucionar un problema primero debemos ser capaces de detectarlo, y aquí es donde entra en juego la importancia de nuestro estudio. Según las conclusiones extraídas del VI foro de la ciberseguridad de ISMS Forum Spain [1] el 75% de los ciberataques no son detectados.

A través del constante análisis de flujos y de las conclusiones extraídas por algoritmos de aprendizaje como los árboles de decisión, los cuales utilizaremos en nuestro estudio, seremos capaces de reducir dicha cifra al igual que el riesgo de que nuestros datos más personales caigan en las manos equivocadas. Se utilizará el árbol J48 para clasificar los diferentes flujos entre tráfico normal o tráfico anómalo y se pretende conseguir unos resultados similares a los de la referencia [2].

1.2 Objetivos

El objetivo principal de este TFG consiste en realizar un estudio sobre los datos de entrada, los cuales estarán formados por flujos pertenecientes a tráfico real mezclado con tráfico de ataque generado sintéticamente, y ver qué parámetros son los que más nos ayudan a detectar cada tipo de ataque. Para llegar a esta conclusión se ha seguido una serie de pasos, los cuales son:

- Filtrado inicial de los datos para ajustarlos al formato del programa que nos devolverá los árboles de decisión resultantes.

- Análisis de los parámetros de mayor peso identificados por el programa, justificando la veracidad de dichas conclusiones.

- Realización de pruebas y extracción de conclusiones con el objetivo de maximizar el porcentaje de aciertos del clasificador.

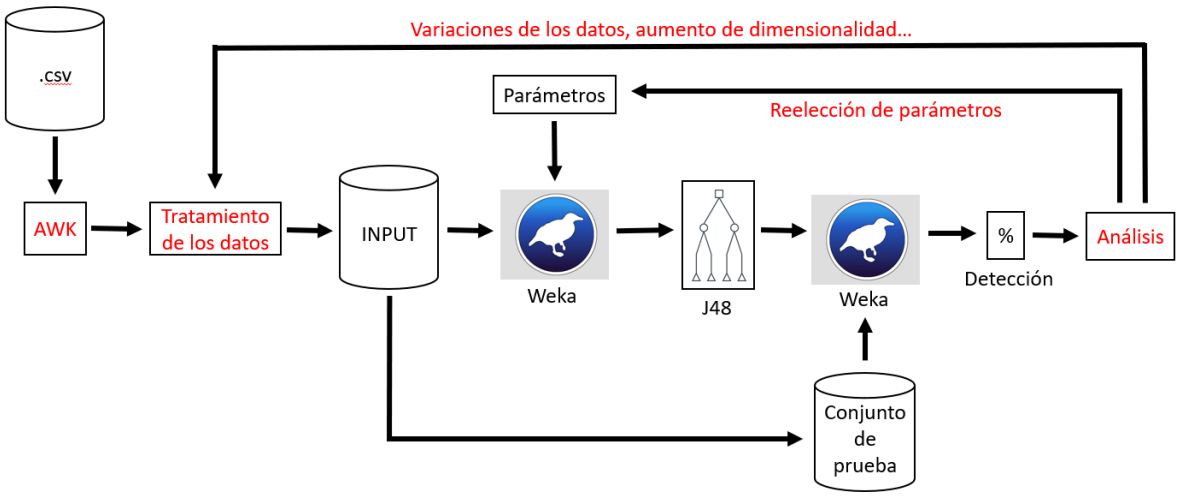


Ilustración 1.2 Esquema del proceso de datos

Las actividades objeto de este Trabajo de Fin de Grado son las que aparecen resaltadas de color rojo.

1.3 Fases de realización

A continuación, se expondrá de manera resumida las diferentes fases que se han realizado para cumplir nuestro objetivo. Dichas fases son:

- **Filtrado de los datos:** Tras identificar el formato necesario de los datos para ser interpretado por Weka, se procede a extraer un conjunto de datos inicial para empezar a realizar las pruebas.

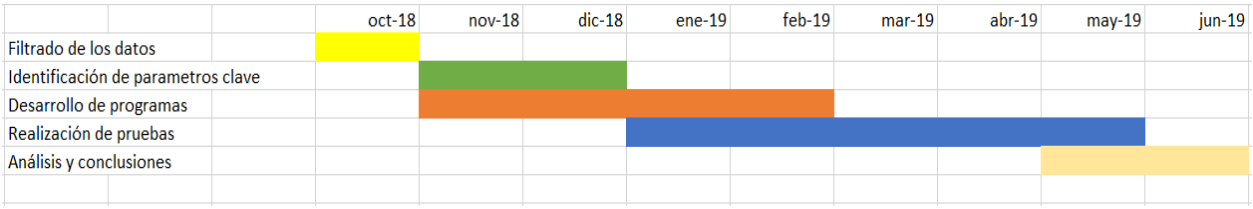
- **Identificación de parámetros clave:** Se realiza un análisis a primera vista de cuáles serán los parámetros que más información nos van a aportar a la hora de realizar la clasificación.

- **Desarrollo de programas:** Con el objetivo de mejorar los resultados obtenidos por los algoritmos, se desarrollan una serie de programas en Python para tratar la gran cantidad de datos y modificarlos en función de las necesidades.

- **Realización de pruebas:** Creando diferentes datasets y escogiendo distintos tipos de ataques se extraen los primeros árboles de decisión, variando los parámetros introducidos para comprobar la efectividad de cada uno de ellos.

- **Análisis y conclusiones:** Después de obtener los árboles de decisión que mejor clasifican cada tipo de ataque, se analizan los distintos parámetros utilizados en cada uno de ellos para identificar el sentido de la elección por parte del algoritmo.

A continuación, se muestra un diagrama de Gantt correspondiente a las fases de realización del estudio:



1.4 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1: introducción.** Conceptos generales
- **Capítulo 2: estado del arte.** En este capítulo se introducirán una serie de conceptos con el objetivo de ponernos en contexto y de definir las herramientas que se utilizarán.
- **Capítulo 3: diseño y desarrollo.** Se expondrán las decisiones de diseño tomadas para mejorar el porcentaje de aciertos a la hora de clasificar los flujos.
- **Capítulo 4: pruebas y resultados.** Se explicarán las diferentes pruebas realizadas y los resultados obtenidos con el objetivo de verificar que la toma de decisiones realizada en el capítulo anterior era correcta.
- **Capítulo 5: conclusiones y trabajo futuro.** A partir de los resultados obtenidos en el capítulo anterior se realizará un análisis y se comprobarán las hipótesis iniciales. Además, se comentarán posibles mejoras para aumentar el alcance de nuestro estudio.

2 Estado del arte

2.1 Introducción

En este capítulo se introducirán diferentes conceptos que están altamente relacionados con nuestro estudio. La base de estos conocimientos es lo que nos va a permitir desarrollarlos más a fondo para poder utilizarlos a nuestro favor y determinar conclusiones nuevas a partir de los datos proporcionados. Esta sección se divide en:

- Ataques de red
- Datasets
- Flujos de red
- Árboles de decisión

2.2 Ataques de red

Hoy en día, Internet es una de las herramientas más utilizadas para muchas de las acciones que realizamos día a día. Lo utilizamos tanto para trabajar, como para descansar, disfrutar o aprender. Damos nuestros datos personales sin pensárnoslo dos veces, descargamos ficheros sin realizar ningún análisis previo, abrimos páginas inseguras… Todas estas acciones aumentan nuestra vulnerabilidad ante los peligros que acechan en la red.

Existen usuarios malintencionados que intentan aprovechar todos nuestros despistes o nuestro desconocimiento para su favor. Utilizan distintas técnicas para atacar las redes de ordenadores, inutilizar los servidores donde nos conectamos o invadir nuestra privacidad. Los expertos en seguridad informática se encargan de velar por nosotros y de construir sistemas que sean lo más inmunes posibles a dichos ataques. Muchos de los ataques se ejecutan gracias a la interacción del usuario, por lo que es de vital importancia conseguir concienciar a la población y dotarles de unos conocimientos básicos con respecto a los peligros de internet. Existen diferentes tipos de ataques en función de lo que se quiere conseguir:

- Software malicioso o “malware” en Internet: Cuando un usuario navega por la web puede infectarse por un malware sin darse cuenta. Los tres tipos más conocidos de malware son los virus, troyanos y gusanos.
  Una vez que nuestro equipo ha quedado infectado el atacante puede realizar multitud de acciones como, por ejemplo, borrar datos, recopilar información privada, ver nuestra pantalla e incluso recibir todo lo que escribimos con nuestro teclado.
  Se suelen expandir de forma autoreplicada, buscando conexiones con otros equipos dentro de una red para así causar los mayores estragos posibles. Uno de los casos más famosos y recientes de este tipo de ataque es el del WannaCry, que es un tipo de Ransomware que se encargaba de cifrar todos nuestros archivos del equipo para después pedir una recompensa económica por el descifrado.
  Existen situaciones en las que podemos estar siendo parte de un ataque sin que nos perjudique directamente. Este sería el caso de que nuestro ordenador esté siendo utilizado para participar en una red botnet que construyen los atacantes

para hacer un ataque de Denegación de Servicio Distribuido (DDoS), el cual explicaremos en la sección 3.3.

- Ataques a servidores e infraestructuras de red: Estos ataques están relacionados con ataques de Denegación de Servicio (DoS), en los cuales se colapsa totalmente un servidor formando un cuello de botella que hace imposible (o muy lenta) la navegación a través de él. Este tipo de ataques se puede frenar utilizando un cortafuegos y restringiendo el acceso a las direcciones IP del atacante. El problema aparece cuando se produce una Denegación de Servicio Distribuido (DDoS) ya que al utilizar más equipos atacantes se nos complica la tarea de bloquear todas las direcciones IP, y el servidor termina por saturarse. En ese caso es difícil diferenciar a un usuario legítimo de un atacante, por lo que será mucho más complicado defenderse de estos.

- Análisis de los paquetes que fluyen por la red ("sniffers"): Actualmente la manera más común de conectarnos a Internet es por medio de las redes WiFi. Un sniffer es un programa que se encarga de analizar los paquetes que se encuentran en una red, por ejemplo, WireShark.
  Si realizamos un ataque Man in The Middle, el cual consiste en interceptar el tráfico que fluye por la red y modificarlo o simplemente observarlo con fines maliciosos. Al conectarnos a una red WiFi doméstica todos los paquetes fluyen por el aire, pero si usas clave estarán cifrados. El problema reside en la habilidad del atacante de crackear la clave de cifrado para acceder al tráfico de la red.
  El sniffing no se produce solo en redes inalámbricas, en las redes LAN cableadas también ya que los paquetes se difunden por cable ethernet, y por tanto pueden ser analizados y modificados.

- Suplantación de identidad: Se suele producir cuando el atacante modifica datos del paquete original para reenviarlo a la red. Puede inyectar paquetes con una dirección IP de origen falsa para así camuflar su identidad real. El router del receptor puede ejecutar el paquete y sin saberlo ejecutaría un comando para modificar la tabla de reenvío. Para solucionar este problema precisamos de mecanismos o procedimientos para poder autentificar los orígenes de las conexiones.

Esta información ha sido extraída de [3].

2.3 Datasets

Un dataset, denominado en castellano como un conjunto de datos, es como dice su nombre un agrupamiento de datos que mantienen una relación y que compone una tabla de una base de datos. Cada columna corresponde a una variable en particular, y cada fila representa un miembro determinado del conjunto de datos en cuestión. Conjuntos de datos tan grandes que no se pueden procesar a través de las aplicaciones tradicionales se denominan big data. [4]

Los datasets son muy utilizados en todo tipo de estudios y análisis ya que nos permiten sacar conclusiones nuevas a partir de un conjunto de datos creado con anterioridad.

Jugarán un papel muy importante en los algoritmos de predicción y en todo lo que tenga que ver con la inteligencia artificial, ya que estos algoritmos suelen requerir de una fase de entrenamiento en la que se extraen conclusiones a partir del dataset introducido. Será de gran importancia asegurarnos de que el conjunto de datos ha sido generado de manera correcta, ya que de este dependerán todos nuestros resultados.

En nuestro caso, utilizaremos un dataset generado por la Universidad de Granada [5] que ha sido creado específicamente para evaluar sistemas de detección de intrusiones, lo cual es el objetivo principal de nuestro estudio. Los datos se han obtenido de una red real de un ISP Tier3. El ISP es un proveedor de servicios en la nube. Esta red se utiliza por muchas compañías que se centran en una gran variedad de mercados, por lo que se espera que el tráfico que atraviesa la red sea muy heterogéneo. Esta es una gran ventaja de dicha traza sobre otras, ya que representa un alto subconjunto de usuarios de Internet. El dataset completo contiene dos capturas distintas:

- **Conjunto de calibración:** captura de flujos con una duración de 100 días. Su principal propósito es ayudar a la construcción y calibración de modelos de normalidad, principalmente porque no se generaron ataques de forma artificial. Tiene un tamaño mucho mayor que el conjunto de prueba.

- **Conjunto de prueba:** captura de flujos con una duración aproximada de un mes. Este conjunto pretende ser utilizada para la validación de algoritmos de detección.

Al tráfico real capturado en la red se le ha añadido ataques generados sintéticamente para posteriormente evaluar el tráfico con le objetivo de poder clasificar entre flujos normales y anómalos provenientes de un ataque. Se instalan 25 máquinas virtuales con una configuración similar a las proporcionadas para los clientes ISP. Las máquinas virtuales se utilizan para lanzar ataques específicos a lo largo del tiempo. A continuación, se muestra la cantidad de flujos de distintos tipos de tráfico a lo largo del tiempo que dura la captura:

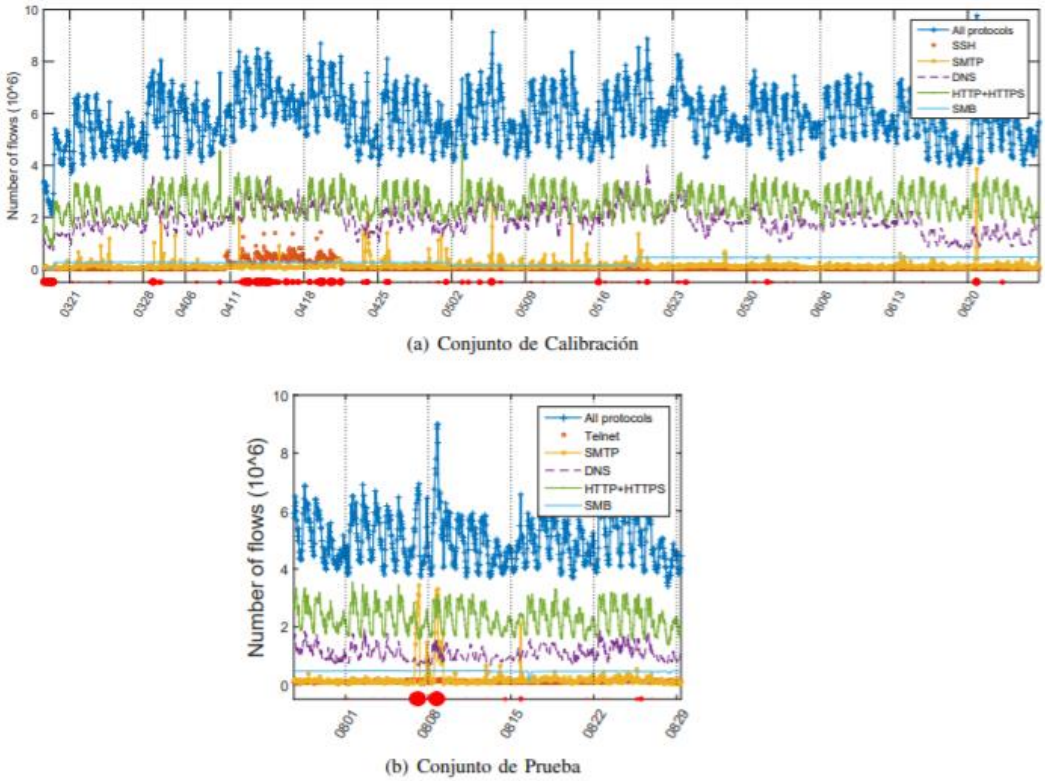


Ilustración 2.1 Evolución del número de flujos para (a) conjunto de Calibración y (b) conjunto de Prueba.

2.4 Flujos de red

Podemos definir los flujos de red [6] como el conjunto de paquetes sucesivos que comparten protocolo, direcciones IP y puertos (origen y destino). Respecto a esta definición genérica se pueden añadir más campos como ocurre en nuestro dataset elegido para realizar el estudio. Este conjunto de atributos permite identificar cada flujo de forma única. La tecnología de NetFlow nos permite realizar tareas como la monitorización, la predicción de ataques o la detección de intrusos. Las cuales serán bastante importantes a la hora de realizar nuestro estudio. Nos centraremos sobre todo en la detección de tráfico anómalo para identificar la presencia de un ataque.

Existen ciertas características presentes en nuestro dataset que nos pueden facilitar el proceso de encaminamiento aportándonos información de interés como, por ejemplo, los tiempos de inicio y fin del flujo, tamaño del paquete, flags activos…

En general, se considera que un flujo ha terminado cuando no se observa tráfico durante 15 segundos, cuando está activo durante más de 30 minutos, cuando recibimos una bandera de fin de conexión o cuando el router se queda sin recursos.
Para evitar limitaciones causadas por el router o la falta de memoria interna se analizan solo un porcentaje de los paquetes para así reducir la carga de trabajo. Tras el submuestreo los datos no podrán ser recuperados de forma exacta pero la precisión perdida no es de gran relevancia en muchas métricas.

2.5 Árboles de decisión

Un árbol de decisión [7] es un modelo de predicción utilizado sobre todo en el campo de la inteligencia artificial. Dado un conjunto de datos se fabrican diagramas de construcciones lógicas que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema. Se suelen utilizar en modelos de predicción para anticipar matemáticamente, en función de los datos proporcionados, la respuesta con mayor porcentaje de probabilidad. En nuestro caso serán la parte fundamental de nuestro estudio y el elemento que más valor va a aportar a la hora de tomar decisiones en función de qué parámetros son los que mejor ayudan a clasificar el tráfico.
Por lo general, un árbol de decisión comienza con un único nodo y luego se ramifica en resultados posibles. A medida que aumenta el número de nodos, aumenta el número de posibles finales a los que puede llegar el individuo.

Los árboles de decisión serán útiles para nuestro estudio por varios motivos:

- Nos permiten manejar tanto atributos continuos como discretos. Crean un umbral y luego dividen la lista en aquellos cuyo valor de atributo es superior al umbral y los que son menores o iguales a él.
- Devuelven un diagrama de árbol muy fácil de entender gracias a su representación visual, lo que nos permitirá identificar todos los posibles caminos y ver en qué punto se etiquetan cada uno de los flujos.
- Permiten manejar atributos con costos diferentes asignando más peso a unos que a otros a la hora de seleccionar los nodos. Gracias a esto podemos obtener un árbol más simplificado que solo mantenga los atributos de mayor peso. Probablemente con este árbol aumentará en número de errores, pero esto puede compensar ya que reduciremos mucho su tamaño sin tener un gran impacto en los resultados. Esta tarea va a ser de gran importancia a la hora de desarrollar nuestro estudio.

Los árboles suelen funcionar a partir de un modelo matemático basado en cálculos de entropías, ganancia de información y las probabilidades de que el atributo en cuestión sea el que mejor clasifica. Podemos encontrar dos tipos de árboles de decisión en la minería de datos [8]:

- **Árboles de clasificación:** Donde la variable de destino puede tomar un conjunto finito de valores. En esta estructura, las hojas representan etiquetas de clase y las ramas representan las conjunciones de características que conducen a esas etiquetas de clase. El resultado predicho es la clase a la que pertenecen los datos.

- **Árboles de regresión:** Donde la variable de destino puede tomar valores continuos (por lo general números reales) como, por ejemplo, el precio de una casa.

Los árboles utilizados para la regresión y los árboles utilizados para la clasificación tienen algunas similitudes pero también algunas diferencias, tales como el procedimiento utilizado para determinar donde dividir. Algunas técnicas construyen más de un árbol de decisión:

- **Bagging:** construye múltiples árboles de decisión haciendo repetidamente remuestreo de los datos de entrenamiento con sustitución, y votando los árboles para hallar una predicción de consenso.
- **Clasificador Random Forest:** utiliza una serie de árboles de decisión, con el fin de mejorar la tasa de clasificación.
- **Los Árboles Impulsados:** se pueden utilizar para problemas de regresión y de clasificación.
- **Rotation Forest:** En el que cada árbol de decisión es entrenado aplicando primero análisis de componentes principales (ACP) en un subconjunto aleatorio de las características de entrada.

2.6 Conclusiones

De este capítulo podemos extraer varios conceptos clave. Los flujos de red son los que contienen la información que vamos a utilizar para sacar las conclusiones de nuestro estudio. Esta información será obtenida gracias a los árboles de decisión, que serán la herramienta principal que utilizaremos. Entre los flujos habrá una mezcla de diferentes ataques generados y flujo normal para así crear un clasificador que nos permita diferenciar entre dichos tipos de tráfico.

3 Diseño y desarrollo

3.1 Introducción

En este capítulo se habla del filtrado realizado a los datos y la toma de decisiones de los parámetros que serán descartados o utilizados en función del ataque en cuestión. También se exponen los motivos por los que se ha escogido el árbol J48 para realizar el estudio. Esta sección se divide en:

- Formato de los datos
- Ataques escogidos
- Weka
- J48
- Programas desarrollados

3.2 Formato de los datos

Durante todo el trabajo se utilizan únicamente datos extraídos de la Universidad de Granada, ya que se pretende sacar conclusiones con sentido para un caso en concreto para poder avanzar en el estudio y poder ir ampliándolo hacia un estudio más general. La base de datos que nos proporciona dicha universidad es muy completa, en la cual encontramos diversos tipos de ataques generados sintéticamente junto a trafico normal recopilado a partir de una red real.

La primera tarea para realizar era entender el formato de los datos junto a cada parámetro en los que estaban divididos los flujos. Los datos eran descargados en un formato .csv donde cada fila representa un flujo y cada columna un parámetro. Los parámetros por orden de aparición son los siguientes:

- Timestamp del final de un flujo: es una secuencia de caracteres que denotan la hora y fecha en la que ocurrió determinado evento. En este caso se corresponde con el inicio del flujo y tiene una resolución de horas y minutos, pero no de segundos. Esto será un factor determinante a la hora de estudiar la importancia de este parámetro, ya que no nos aportará el grado de detalle que precisamos.

- Duración del flujo: es la cantidad de tiempo, medida en segundos, que pasa desde el inicio del flujo hasta su fin.

- Dirección IP de origen y destino: Son las direcciones de la máquina origen y de la máquina destino.

- Puerto origen y puerto destino: Son los puertos a los que va dirigido el flujo y desde donde proviene.

- Protocolo: Se utiliza para nombrar las normativas y los criterios que fijan cómo deben comunicarse los diversos componentes de un cierto sistema de interconexión.

- Banderas: Existen distintos tipos de banderas, pero son exclusivas del protocolo TCP. Su activación nos proporcionará información sobre ciertas condiciones. Las banderas utilizadas en nuestro dataset son las siguientes:

    - SYN (Synchronize): Se utiliza para iniciar una conexión TCP.

    - ACK (Acknowledgement): Se usa para confirmaciones, si en la conexión nos lo devuelven activo significa que el paquete fue recibido con éxito.

    - RST (Reset): Este bit nos permite reiniciar una conexión debido a paquetes corrompidos o a SYN duplicados, retardados…

    - PSH (Push): Se utiliza para forzar el enviado inmediato de los datos tan pronto como sea posible.

    - URG (Urgent): Sirve para definir un bloque de datos como urgente.

    - FIN (Finalize): Finaliza la conexión.

- Estado de reenvío: Si está activo significará que se ha producido un reenvío de los datos hacia otra dirección.

- Tipo de servicio: Indica una serie de parámetros sobre la calidad de servicio deseada durante el tránsito por una red. Algunas redes ofrecen prioridades de servicios, considerando determinados paquetes más importantes que otros.

- Paquetes intercambiados en el flujo: Es el número de paquetes que se han intercambiado dentro de un flujo desde el inicio de este hasta su fin.

- Correspondiente número de bytes: Es el tamaño que ocupa la información que se pretende transmitir más los campos obligatorios en función de la cabecera escogida.

Más adelante se explicará el motivo por el cual se ha prescindido de alguno de los parámetros.

3.2.1 Formato .arff

Weka solo admite ficheros con formato .arff. Estos ficheros con formato .arff se dividen en tres partes [9]:

1. **@relation <relation-name>:** Todo fichero ARFF debe comenzar con esta declaración en su primera línea (no podemos dejar líneas en blanco al principio). <relation-name> será una cadena de caracteres y si contiene espacios la pondremos entre comillas.

2. **@attribute <attribute-name> <datatype>:** En esta sección incluiremos una línea por cada atributo (o columna) que vayamos a incluir en nuestro conjunto de datos, indicando su nombre y el tipo de dato.

3. Con <attribute-name> indicaremos el nombre del atributo, que debe comenzar por una letra y si contiene espacios tendrá que estar entrecomillado. Con <datatype> indicaremos el tipo de dato para este atributo (o columna) que puede ser:

    a. **Numeric** (numérico)

    b. **String** (texto)

    c. **Date [<date-format>]** (fecha). En <date-format> indicaremos el formato de la fecha, que será del tipo "yyyy-MM-dd'T'HH:mm:ss".

    d. **<nominal-specification>:** Estos son tipos de datos definidos por nosotros mismos y que pueden tomar una serie de valores que indicamos.

4. **@data:** En esta sección incluiremos los datos propiamente dichos. Separaremos cada columna por comas y todas filas deberán tener el mismo número de columnas, número que coincide con el de declaraciones @attribute que añadimos en la sección anterior. Si no disponemos de algún dato, colocaremos un signo de interrogación (?) en su lugar. El separador de decimales tiene que ser obligatoriamente el punto y las cadenas de tipo string tienen que estar entre comillas simples.

Utilizando una de las herramientas que nos proporciona el programa podríamos pasar de un fichero en formato .csv a uno en formato .arff. Esta herramienta se conoce como ArffViewer. Podemos observar un volcado de la herramienta a continuación:

Relation: outputaleatorio(dos)

No.	1: duracion	2: iporigin	3: ipdest	4: portorigin	5: portdest	6: protocol	7: flagU	8: flagA	9: flagP	10: flagR	11: flagS	12: flagF	13: estadofw	14: servicio	15: paqinter	16: numbytes
	Numeric	Nominal	Nominal	Numeric	Numeric	Nominal	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
1	0.104	78.160...	42.21...	443.0	26375.0	TCP	0.0	1.0	1.0	0.0	0.0	0.0	0.0	40.0	2.0	141.0
2	0.244	42.219...	78.16...	28460.0	443.0	TCP	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	2.0	141.0
3	0.38	42.219...	68.11...	28266.0	443.0	TCP	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	3.0	194.0
4	0.404	42.219...	116.2...	35920.0	80.0	TCP	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	2.0	116.0
5	10.0	42.219...	122.2...	47.0	123.0	UDP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	152.0
6	10.0	42.219...	60.62...	47.0	123.0	UDP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	152.0
7	0.468	42.219...	133.5...	80.0	51404.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	6.0	5536.0
8	0.476	42.219...	43.16...	26754.0	443.0	TCP	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	24.0	6842.0
9	0.688	105.16...	42.21...	5222.0	27147.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	72.0	8.0	751.0
10	0.76	42.219...	167.1...	80.0	43285.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	7.0	7486.0
11	0.828	42.219...	105.1...	27147.0	5222.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	10.0	838.0
12	1.056	42.219...	68.11...	30998.0	5222.0	TCP	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	3.0	194.0
13	10.92	43.164...	42.21...	443.0	58186.0	UDP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	120.0	129344.0
14	11.396	43.164...	42.21...	80.0	59185.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	16.0	9469.0
15	1.224	42.219...	214.1...	65467.0	445.0	TCP	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	3.0	144.0
16	1.368	213.46...	42.21...	14530.0	52235.0	TCP	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	120.0
17	1.544	42.219...	212.7...	143.0	62606.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	18.0	3606.0
18	1.648	212.76...	42.21...	62606.0	143.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	23.0	1577.0
19	19.964	42.219...	122.2...	28.0	123.0	UDP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	228.0
20	29.992	42.219...	60.62...	28.0	123.0	UDP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	304.0
21	70.08	145.23...	42.21...	443.0	49560.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	74.0	31165.0
22	1.052	42.219...	223.8...	80.0	55762.0	TCP	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	2.0	120.0
23	1.056	42.219...	223.8...	80.0	55760.0	TCP	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	2.0	120.0
24	1.12	42.219...	223.8...	80.0	55719.0	TCP	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	2.0	120.0
25	1.212	42.219...	223.8...	80.0	55795.0	TCP	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	2.0	120.0
26	1.336	42.219...	85.19...	43098.0	443.0	TCP	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	2.0	717.0
27	1.452	42.219...	244.1...	53871.0	443.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	20.0	2390.0
28	1.484	42.219...	217.1...	6881.0	11148.0	UDP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	260.0
29	1.608	78.160...	42.21...	443.0	41898.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	40.0	9.0	5294.0
30	1.644	42.219...	43.16...	55733.0	80.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	8.0	991.0
31	1.696	42.219...	78.16...	41898.0	443.0	TCP	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	11.0	1393.0
32	1.904	42.219...	50.28...	52953.0	55765.0	TCP	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	3.0	152.0
33	20.16	42.219...	84.21...	16716.0	161.0	UDP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	321.0
34	0.0	42.219...	42.21...	2198.0	80.0	TCP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	200.0
35	20.16	42.219...	84.21...	39245.0	161.0	UDP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	321.0
36	21.124	42.219...	115.2...	80.0	60858.0	TCP	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	6.0	3220.0
37	2.932	42.219...	56.18...	60384.0	445.0	TCP	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2.0	96.0
38	2.952	42.219...	49.14...	14542.0	445.0	TCP	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2.0	96.0

Ilustración 3.1 Volcado de la herramienta ArffViewer

3.2.2 Parámetros de interés

Tras analizar los parámetros nos fijamos en el timestamp. Este suele ser un campo importante a la hora de analizar ataques como el de denegación de servicio (DoS) o el escaneo de puertos. En nuestro caso este timestamp tenía una resolución de minutos, por lo que no podíamos verificar el segundo en el que era ejecutado el ataque. Esto era una desventaja a la hora de analizar los datos ya que aproximadamente un minuto equivalía a un conjunto de 300 flujos. Por eso, decidimos eliminar este campo de nuestra base de datos, ya que no aportaría ningún valor a la hora de diferenciar entre tráfico normal y tráfico de ataque. Más adelante se estudió la posibilidad de mantener un timestamp incremental para poder sacar mejores conclusiones. Esto se hallaría restando cada timestamp con el timestamp del flujo anterior, pero ya que tendríamos aproximadamente 300 flujos por minuto, obtendríamos un 1 cada 300 flujos siendo el resto 0's. Esto no nos proporcionaba ninguna información de interés a la hora de procesar los datos.

Otro de los campos que nos traerían problemas a la hora de utilizarlos en el árbol de clasificación serían las direcciones IP de origen y destino. Esto se debe a que si el árbol tomase decisiones en función de dichos campos el estudio sería demasiado especifico y si introdujésemos datos provenientes de otro origen devolvería resultados erróneos. Otra de las razones es que los árboles resultantes eran de un tamaño muy grande debido a que terminaba clasificando los datos en tráfico normal o anómalo en función de las direcciones IP de origen y de destino, lo cual no interesaba a la hora de sacar conclusiones. De todas formas, dichos parámetros se mantendrían en la fase de procesamiento, ya que eran de vital importancia para el agrupamiento de los flujos, pero se eliminarían a la hora de enviarlos al árbol de decisión.

El campo de las banderas es de vital importancia para la realización del estudio, el inconveniente es que venían en un formato específico, ya que si la bandera no estaba activada en dicho flujo vendría representado por un punto y si estaba activada se representaría como la primera letra de dicha bandera. Para poder analizar este campo se transformó dividendo cada una de las posibles banderas activas en un parámetro, y asignándole un 1 o un 0 dependiendo del estado en el que se encontrase. A parte, el dataset presentaba un pequeño error, ya que flujos provenientes de protocolos diferentes al TCP presentaban algún flag activo. Como bien sabemos, el único protocolo que presenta flags es TCP, por lo que al procesar los datos se pondrían todos los flags a 0 si el flujo no utilizaba dicho protocolo. Estas modificaciones se añadirían en los programas desarrollados, los cuales se explicarán en la siguiente sección.

En cuanto al resto de parámetros, a primera vista podrían aportar valor a la hora de cumplir con nuestro objetivo. A medida que se avanza con el estudio y en función de las conclusiones obtenidas, se volvería a replantear la idea de si eran necesarios todos los parámetros o si había alguno que nos perjudicase más que beneficiarnos.

3.3 Ataques escogidos

En el mundo existen infinidad de ataques cibernéticos conocidos y otros muchos desconocidos o en desarrollo. Para comenzar con el estudio, necesitaremos elegir ataques que sean bien conocidos por nosotros y de los cuales podamos extraer conclusiones reales y fiables. Esto es de gran importancia, ya que los caminos de decisión generados por los

algoritmos serán diferentes en función de la procedencia del ataque. Quedará a nuestro juicio decidir si los atributos escogidos para filtrar los datos, sobre todo en la cabeza del árbol (ya que son las primeras decisiones que tomar por el clasificador y es donde se descartarán como anómalos más del 90% de los flujos), son los que mejor se ajustan a cada tipo de ataque y nos proporcionan la mayor probabilidad de aciertos. Por esto, todas las decisiones quedarán razonadas y argumentadas con el objetivo de cumplir lo anterior.

El dataset generado por la Universidad de Granada contiene diversos tipos de ataques, de los cuales hemos escogido una serie de ellos en base a los siguientes criterios:

- Baja especificidad del ataque, ya que no realizan procesos muy complejos para evitar cualquier probabilidad de ser detectados y son conocidos a nivel mundial por ser la base de ataques más avanzados y complejos.
- Alto número de flujos. Hay ciertos ataques de los cuales se han obtenido muy pocos flujos, lo que dificulta el proceso de entrenamiento de un clasificador. Cuanto mayor sea el número de flujos, provenientes de un único ataque, que tenemos a nuestra disposición y mayor sea la variedad entre estos, mayor serán las probabilidades de detectar un ataque proveniente de un dataset diferente.
- Información general conocida sobre los ataques. Esto es de gran importancia, ya que ataques que se ejecuten de una manera totalmente aleatoria realizando acciones diferentes en cada iteración serán muy difíciles de clasificar, y quedarán fuera del alcance de nuestro estudio.

A continuación, se expondrán las características y la funcionalidad que presentan los ataques seleccionados:

- Denegación de servicio (DoS): Es un ataque a un sistema o red de computadores que causa que un servicio que ofrecen o un recurso quede inaccesible para los usuarios legítimos debido a la saturación de los puertos que provoca una sobrecarga del servidor. Esta saturación se suele realizar por medio del envío de múltiples flujos de información desde una misma máquina o dirección IP hacia un mismo puerto destino. El servidor no da abasto a la cantidad de solicitudes y se genera un cuello de botella en el sistema que termina en la caída del servidor. Existe una variante denominada denegación de servicio distribuido (DDoS) que consiste en realizar un ataque DoS, pero empleando un gran número de ordenadores o direcciones IP para así aumentar la cantidad de tráfico exponencialmente y que sea más difícil identificar su origen y bloquear las direcciones IP.
  Dentro de los ataques DoS se pueden producir ataques síncronos o asíncronos. En los primeros todos los ataques se inician por los atacantes al mismo tiempo. En cambio, en el segundo caso se pueden alternar periodos de tiempo en los que se están produciendo ataques a víctimas seleccionadas secuencialmente y periodos de inactividad en los que no se envían flujos. Este será otro factor determinante a la hora de elegir los atributos para realizar nuestro estudio, ya que en ataques asíncronos el campo del timestamp no será de utilidad ya que no podemos predecir la duración de cada uno de los intervalos, y podría variar de manera aleatoria con el tiempo.

- Escaneo de puertos: Esta técnica se emplea para analizar por medio de un programa el estado de los puertos (abierto o cerrado) de una máquina conectada a una red de comunicaciones. Determinados números de puertos se suelen asociar a una serie de

servicios ofrecidos por lo que nos permite detectar los servicios que se están ofreciendo y posibles vulnerabilidades de seguridad según los puertos abiertos. También podemos llegar a detectar el sistema operativo que se está ejecutando en función de los puertos abiertos. En nuestro estudio vamos a analizar tres tipos de escaneo de puertos:

  - Escaneo de puertos UDP: En los rastreos de puertos TCP se suele utilizar paquetes SYN para determinar el estado del puerto en función de los paquetes devueltos. En cambio, el protocolo UDP no cuenta con paquetes SYN, por lo que se utilizan mensajes ICMP para determinar si un puerto está cerrado. La mayoría de los escáneres utilizan dicho método y asumen que, si no se recibe una respuesta, el puerto está abierto.

  - Scan11: Esta técnica ha sido desarrollada por la Universidad de Granada a la hora de crear los flujos de ataque de manera sintética. Consiste en la utilización de paquetes SYN para el escaneo de los puertos más comunes de las víctimas. Se realiza un escaneo uno-a-uno donde un único atacante escanea una víctima.

  - Scan44: Es una técnica similar al Scan11 mencionado anteriormente, pero la diferencia es que se realiza un ataque de escaneo cuatro-a-cuatro donde cuatro atacantes inician al mismo tiempo un escaneo a cuatro víctimas. Los ataques se llevan a cabo en paralelo por lo que no aumentará el tiempo de duración del ataque.

- Actividad relacionada con una botnet: Se simula tráfico de botnet mediante la exfiltración de datos desde algunas máquinas infectadas al puerto 80 de un botmaster. Se utilizarán veinte bots, correspondientes a todas las máquinas víctima. Estos bots realizarán tareas de exfiltración como, por ejemplo, enviar fragmentos de 1KB o 1MB al botmaster en una única conexión o enviar un total de información de 1 MB dividido en fragmentos de 1KB enviados al botmaster en conexiones distintas. Los ataques pueden ser síncronos o asíncronos dependiendo de si los bots inician la transmisión de información al mismo tiempo o cada uno de ellos selecciona un instante aleatorio, respectivamente. Esta es una de las razones por las que no es de utilidad analizar la información incluyendo el campo de timestamp, ya que el momento de inicio de los ataques puede estar distribuido de forma aleatoria.

3.4 Herramienta escogida para el estudio: Weka

Weka es una plataforma de software para el aprendizaje automático y la minería de datos escrito en Java y desarrollado en la Universidad de Waikato [10]. Presenta distintas características que nos pueden aportar valor a la hora de realizar el estudio:

- Está disponible libremente bajo la licencia pública general de GNU.

- Es muy portable porque está completamente implementado en Java y puede correr en casi cualquier plataforma.

- Contiene una extensa colección de técnicas para preprocesamiento de datos y modelado.

Estas características son muy importantes, ya que nos permitirán evaluar los datasets con diferentes algoritmos para medir su eficacia y eficiencia. A parte, Weka es una herramienta muy útil para la detección de intrusiones de red debido a que tiene implementado un árbol de decisión muy peculiar que nos aportará una gran exactitud a la hora de clasificar flujos de red. Este árbol es el J48 y en la siguiente sección se comentarán las razones por las que se eligió dicho algoritmo.

3.5 Árbol escogido para el estudio: J48

C4.5 es un sucesor de ID3 desarrollado por Ross Quinlan y se implementa en Weka como J48 usando Java [11]. Todos ellos adoptan un enfoque de arriba hacia abajo para la toma de decisiones. Los nodos con más peso serán los que aparecen en la cima del árbol, ya que serán los que dividan la mayor cantidad de flujos entre sus dos caminos. Este árbol se utiliza para la clasificación en la que los nuevos datos se etiquetan de acuerdo con observaciones ya existentes. Además de un conjunto de datos también se envía un conjunto de atributos. Cada flujo en el conjunto de datos está asociado a una etiqueta de clase que identifica si proviene de tráfico normal (background) o de tráfico de ataque (hacked). La división del conjunto de datos y la asignación de la etiqueta correspondiente se realiza a partir de una serie de procesos que seleccionan los atributos que mejor particionan el conjunto de datos. Estas medidas de selección de atributos son responsables para el tipo de ramificación que se produce en un nodo. El procedimiento que sigue este algoritmo para elegir el atributo para cada nodo es el siguiente:

1. Cada atributo tiene una ganancia de información asociada y se puede definir como la reducción de la entropía causada por la división de las instancias en función de los valores obtenidos por ese atributo. La ganancia de información se calcula a partir de la siguiente fórmula:

$$Ganancia\ de\ información(N,A) = Entropía(N) - \sum_{valores(A)} \frac{|N_i|}{|N|} Entropía(N)$$

Ilustración 3.2 Fórmula de cálculo de la Ganancia de Información.

Donde N es el conjunto de instancias en ese nodo en particular y Ni es el subconjunto de N para el cual el atributo A tiene valor i. La entropía del conjunto N se calcula como:

$$Entropía(N) = \sum_{i=1}^{N^{o}\ de\ clases} -P_i \log_2(P_i)$$

Ilustración 3.3 Fórmula de cálculo de la Entropía.

Donde Pi es la proporción de instancias en N que tienen su i-ésimo valor de clase como atributos de salida. Para un ejemplo en el que haya 2 clases, p2 será la fracción de ejemplos positivos y p1 la fracción de negativos.

2. Se calcula la ganancia de información de cada nodo y se elegirá el mayor valor de dicha ganancia para continuar con la clasificación.

Cuando se realiza la asignación del atributo correspondiente siguiendo los pasos mencionados anteriormente, se agregan nuevas ramas para cada valor tomado por el atributo de prueba debajo de sus respectivos nodos. En cada nodo, las instancias de entrenamiento pasan por la rama junto con su valor de atributo de prueba asociado y, además, este subconjunto de instancias de entrenamiento se usa recursivamente para crear nuevos nodos. Si el valor de salida no varía, a continuación, se genera una hoja para finalizar la recursión de nodos en la rama y se le asigna al atributo de salida el mismo valor de clase.

Como conclusión, se han realizado diferentes pruebas para la detección de intrusiones con varios algoritmos que ofrece la herramienta Weka. Tras ello podemos observar que el algoritmo que devuelve los mayores porcentajes de acierto para la realización de nuestro estudio es el J48 con valores cercanos al 100%, frente a algoritmos como Naïve Bayes que presentan valores por debajo del 90% [2].

3.6 Programas desarrollados

Para variar la estructura de los datos y con el objetivo de reducir las probabilidades de error del árbol de decisión se desarrollaron una serie de programas que permitiesen procesar y modificar un gran volumen de datos.

El primer programa, desarrollado en Python, fue creado con la intención de aumentar la dimensionalidad de los datos. El objetivo era agrupar los flujos de datos que proviniesen de una misma conexión origen-destino para posteriormente unirlos formando una única fila, la cual Weka percibiría como un único flujo, pero con muchos más parámetros. Se tomarían flujos de 4 en 4 y se iría aumentando dicha dimensionalidad para observar si efectivamente aumentaba el número de aciertos a medida que incrementábamos dicho parámetro.

El segundo programa fue desarrollado también en Python. El objetivo era observar si los árboles variaban en función de cómo presentábamos los datos. En las primeras pruebas separábamos el tráfico normal del tráfico de ataque totalmente, al principio del fichero poníamos uno y al final otro. Como esta no es una distribución típica del tráfico observado en una red real queríamos ajustarnos lo máximo posible a la realidad para así conseguir unos resultados más fiables, y extrapolables a otros datasets. La alternativa era, a medida que se procesaba el fichero, ir añadiendo flujos de ataque en función de una probabilidad determinada. Suponiendo una presencia de tráfico en la red del 1% del total, tomaríamos esa misma probabilidad para introducir los flujos malignos. De esta forma conseguimos obtener una distribución mucho más cercana a la realidad.

3.7 Conclusiones

En este capítulo se presenta por primera vez el dataset que va a ser utilizado para realizar el estudio, justificando cuáles son los parámetros que más peso o menos pueden tener a la hora de clasificar los ataques. Se definen todos los ataques utilizados y las características más importantes de cada uno que serán útiles a la hora de sacar conclusiones. Se explica de manera detallada el funcionamiento del árbol elegido en cuestión y los programas desarrollados para facilitar el tratamiento de los datos (ya que estos son de gran tamaño) y mejorar la eficacia de los resultados.
Podemos identificar tres tareas principales: filtrado de los datos, para poder operar con ellos en la herramienta Weka, análisis de los parámetros, ya que estos son los más importantes a la hora de construir el árbol de decisión e identificar el tipo de algoritmo que mayor probabilidad de acierto nos proporcione.

A continuación, en el capítulo 4 se comentarán las pruebas realizadas y las conclusiones extraídas de los árboles de decisión obtenidos para cada ataque.

4 Pruebas y resultados

4.1 Introducción

En este capítulo se presentarán las pruebas realizadas para mejorar el desempeño del árbol de decisión y para comprobar la hipótesis inicial. Se comentarán los resultados obtenidos para cada tipo de ataque. Esta sección se divide en:

- Aumento de la dimensionalidad
- Inyección de tráfico normal a un dataset con varios ataques
- Extracción del árbol de decisión para cada tipo de ataque

4.2 Aumento de la dimensionalidad

Gracias a la ayuda del programa que hemos desarrollado para aumentar la dimensionalidad podemos comprobar si esta acción es beneficiosa para nuestro estudio o no. Nos fijaremos en la probabilidad de acierto asociada a cada aumento de la dimensionalidad de los datos. Realizaremos distintas pruebas:

- Los datos sin modificar con la dimensionalidad inicial.
- Aumentaremos la dimensionalidad de los datos añadiendo tres flujos concatenados por línea.
- Aumentaremos la dimensionalidad de los datos añadiendo cuatro flujos concatenados por línea.
- Aumentaremos la dimensionalidad de los datos añadiendo ocho flujos concatenados por línea.

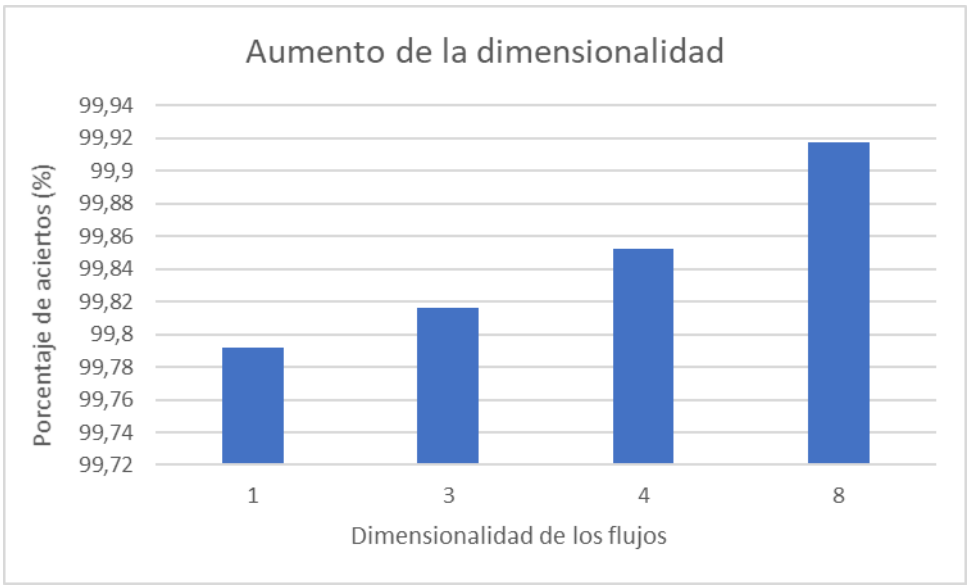


Ilustración 4.1 Gráfica que representa el aumento del porcentaje de aciertos con respecto al aumento de la dimensionalidad de los datos.

Podemos comprobar mediante las herramientas que nos proporciona Weka que la reducción de la dimensionalidad, a partir de la inicial, también trae consecuencias negativas. Si seleccionamos los atributos con más peso e importancia dentro de la clasificación, Weka nos recomienda quedarnos con 2 dentro de los 15 iniciales, para el caso específico de un ataque DoS. Estos atributos son puerto origen y número de bytes del flujo. Realizando un análisis a primera vista no parece tener mucho sentido quedarnos solo con el puerto de origen, ya que generalmente en dichos ataques se suele enviar muchos paquetes a un mismo puerto de destino para así inundarlo. A continuación, para comprobar el descenso en la probabilidad de acierto se procede a realizar una clasificación de los datos teniendo en cuenta solamente los dos atributos mencionados anteriormente y la etiqueta de tráfico normal (background) o de tráfico de ataque.

La probabilidad pasa de un 99,998% cuando tenemos en cuenta todos los atributos iniciales a un 99,9965% cuando usamos unicamente los tres atributos mencionados anteriormente. Esto refuerza la condición de que si aumentamos la dimensionalidad es bastante probable que aumentemos el número de aciertos. En cambio, esta técnica es más efectiva con la presencia de varios ataques, ya que se han realizado pruebas con datasets que contienen un único ataque y se ha visto que el porcentaje de aciertos es tan grande que no merece la pena aumentar la dimensionalidad ya que esto ocasionaría un aumento del tamaño del árbol.

4.3 Inyección de tráfico normal a un dataset con varios ataques

A continuación, se expondrá la evolución de los resultados a partir de un dataset con un número inicial de flujos de tráfico normal (los cuales irán aumentando) y un número fijo de flujos de ataque de tres tipos: DoS, anomaly-spam y anomaly-udpscan. El objetivo de esta prueba es ver como evoluciona el tamaño del árbol y el número de errores en función del tipo de tráfico normal que se va introduciendo. Se han realizado cuatro pruebas, en las tres primeras se ha introducido tráfico normal proveniente de un mismo dataset y en la última prueba se ha introducido tráfico normal con unas características diferentes a los flujos introducidos anteriormente, ya que este tráfico proviene de otro dataset.

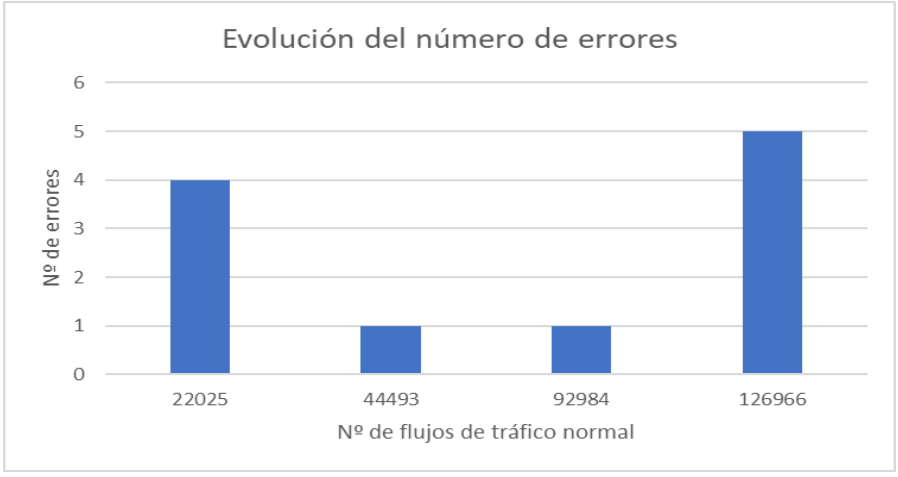


Ilustración 4.2 Gráfica que representa la evolución del número de errores en función del aumento del número de flujos de tráfico normal.

Como podemos observar los errores bajan bastante al aumentar el tráfico normal proveniente del mismo dataset que los flujos iniciales. En cambio, en la última prueba los errores vuelven a subir debido a la variabilidad que presentan los flujos introducidos con respecto a los iniciales.

He creado un gráfico para que podamos ver cómo aumenta la dimensión total del árbol en función del aumento de flujos.

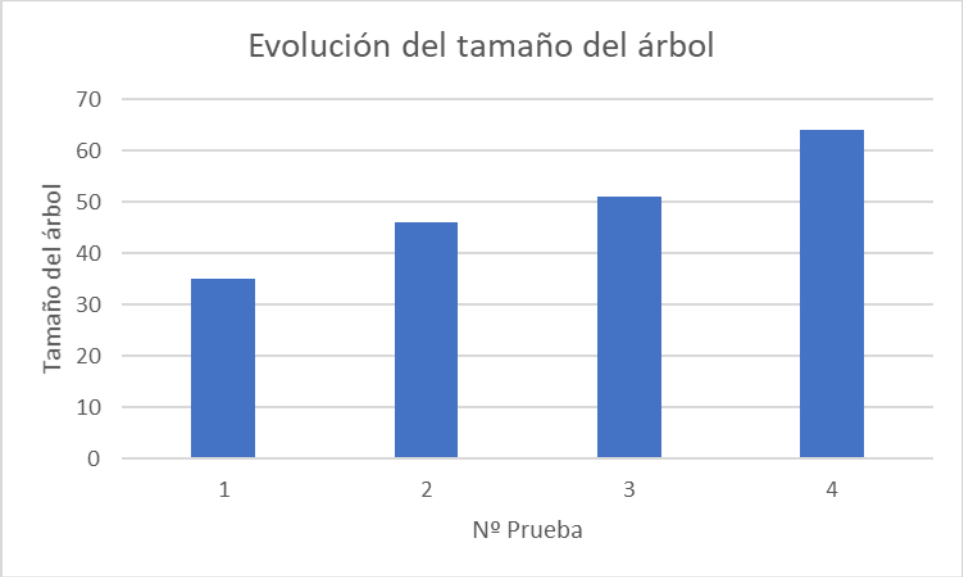


Ilustración 4.3 Gráfica que representa la evolución del tamaño del árbol en función del aumento del número de flujos.

Al aumentar el número de flujos y en presencia de varios ataques el tamaño del árbol tiende a ascender, ya que cada flujo es diferente al otro por lo que necesitamos de más condiciones para poder diferenciarlos del tráfico de ataque. Esto resultará en un mayor número de hojas y por lo tanto el tamaño total del árbol será mayor.

En cuanto a los resultados obtenidos en las matrices de confusión (ver anexo) podemos observar que a medida que introducimos tráfico normal procedente del mismo dataset conseguimos que los errores disminuyan hasta practicamente ser nulos. En cambio, cuando introducimos tráfico normal proveniente de un dataset diferente, estos flujos presentarán unas características diferentes al dataset inicial por lo que los errores suben. De todas formas el aumento es muy pequeño por lo que la conclusión sería que cuantos más flujos tengamos, mejor vamos a poder predecir su origen.

4.4 Extracción del árbol de decisión para cada tipo de ataque

Tras observar los resultados obtenidos en el apartado anterior y fijandonos en el tamaño del árbol nos podemos dar cuenta rapidamente de que un árbol con tantas hojas no va a ser eficiente para datasets provenientes de diferentes origenes. El árbol tiene que ser lo

suficientemente específico, pero a la vez general, posible. Lo ideal sería que tuviese alrededor de cuatro hojas las cuales nos permitirán obtener conclusiones más generales de en qué nos tenemos que fijar a la hora de clasificar tráfico. Para poder conseguir esto debemos dividir los ataques por separado, ya que si tenemos un dataset con varios ataques mezclados, será más dificil diferenciar uno de los otros y del tráfico normal. Es por esto que hemos creado diferentes datasets mezclando tráfico normal y los ataques mencionados en la sección 3. A continuación, representaremos los árboles obtenidos y sacaremos conclusiones sobre los parámetros que ha escogido para cada uno de los ataques.

- Denegación de Servicio (DoS):

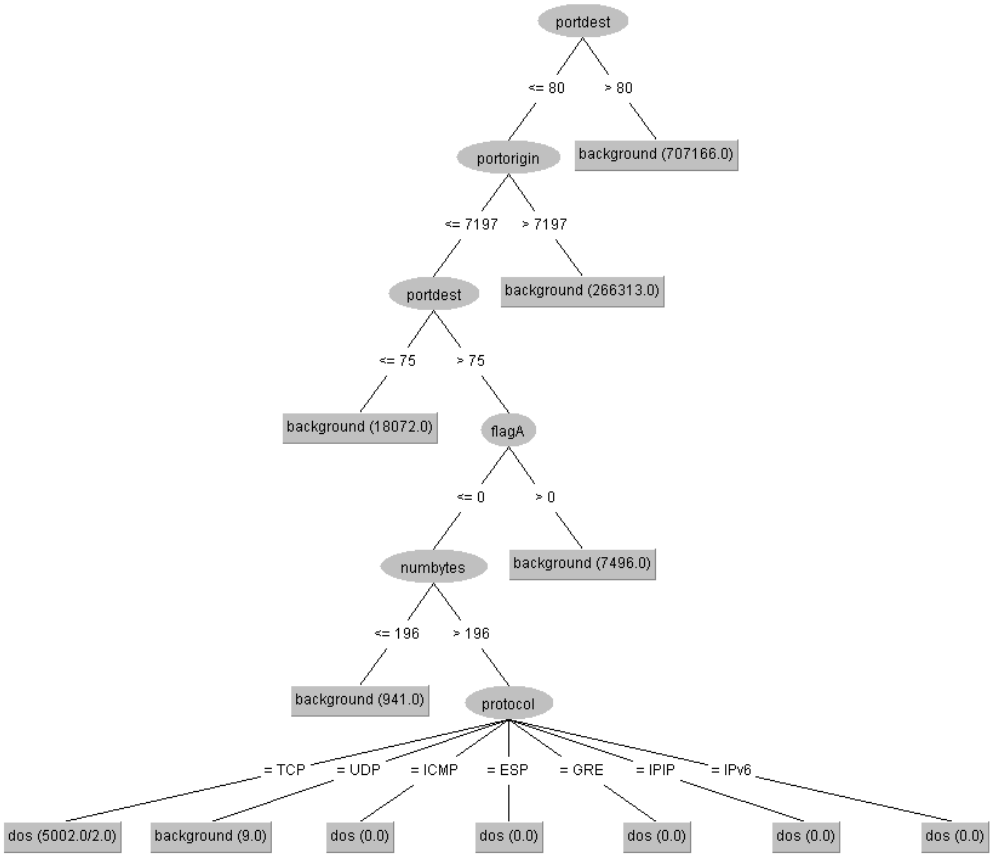


Ilustración 4.4 Árbol completo generado para el ataque DoS.

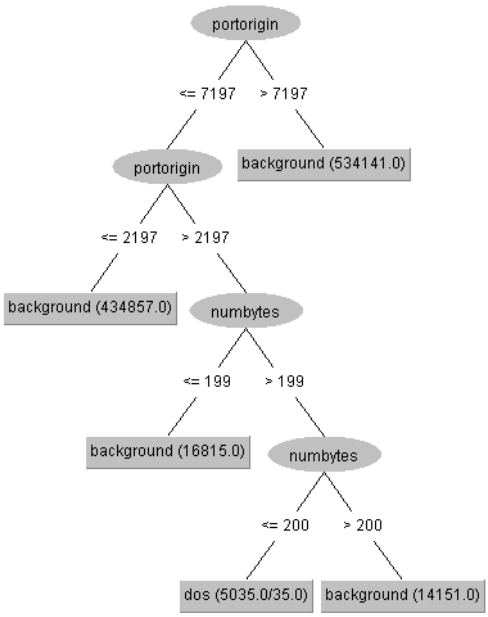


Ilustración 4.5 Árbol reducido generado para el ataque DoS.

	Árbol completo	Árbol reducido
Duración		
Puerto origen	✔	✔
Puerto destino	✔	
Protocolo	✔	
Flag U		
Flag A	✔	
Flag P		
Flag R		
Flag S		
Flag F		
Estado de reenvío		
Tipo de servicio		
Paquetes intercambiados		
Número de bytes	✔	✔

Tabla 4.1 Parámetros utilizados en la construcción del árbol para el ataque DoS

Observamos en la tabla anterior los parámetros de los que depende cada árbol.
Se puede deducir que se está realizando un ataque al servidor web que se encuentra alojado en el puerto 80 por lo que la mayoría de flujos de ataque tienen ese puerto como destino

con la intención de inundarlo. El puerto de origen variará dentro de un rango de diferentes puertos, pero al no ser un DDoS será más sencillo bloquear las direcciones IP y puerto de origen de donde provenga el ataque. El flag ACK es interesante que esté desactivado, ya que en dichas conexiones no es necesario que se confirme la conexión y llegará un punto en el que el servidor objetivo dejará de estar activo por lo que tampoco recibiriamos un ACK de vuelta. El número de bytes puede haber sido predeterminado por el atacante para que todos los flujos tengan un tamaño parecido o aleatorio, pero si enviamos muchos flujos con un tamaño aceptable la máquina objetivo se inundará antes que si enviamos los mismos flujos pero de menor tamaño.

En el caso del árbol reducido no aumentan mucho los errores, pero parece que los parámetros son demasiado específicos como para poder utilizar dicho árbol para otro dataset. Los parámetros seleccionados se pueden considerar muy manipulables por lo que el atacante podrá evadir nuestros patrones de decisión de manera sencilla.

- Escaneo de puertos UDP:

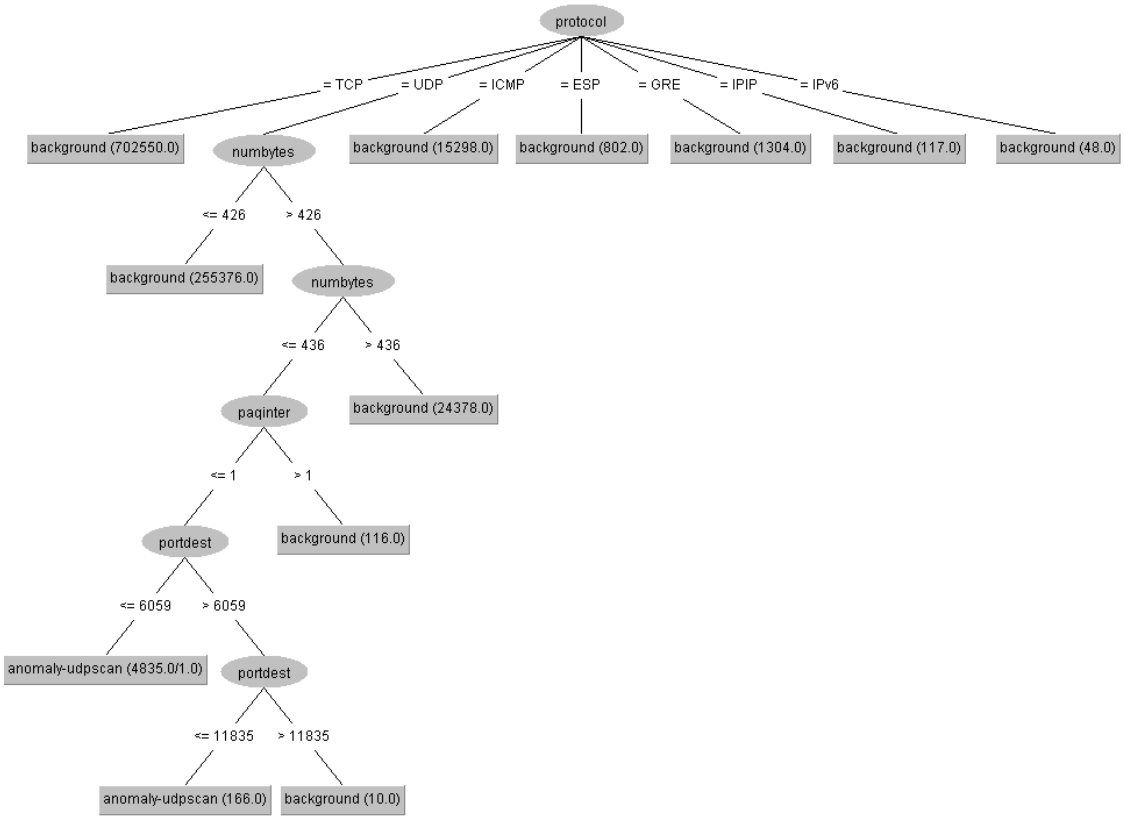


Ilustración 4.6 Árbol completo para el ataque de escaneo de puertos UDP.

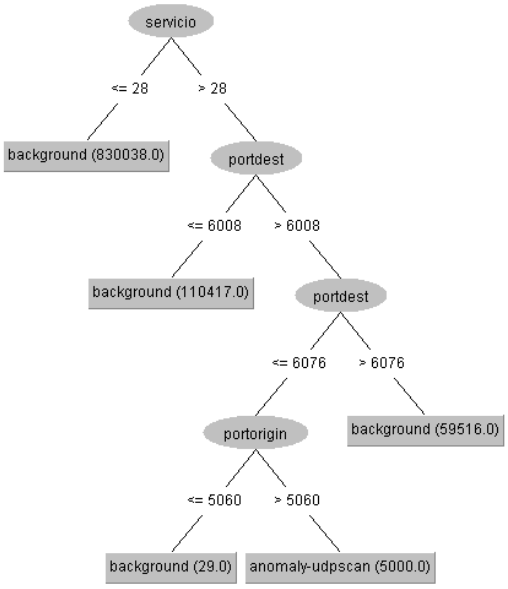


Ilustración 4.7 Árbol reducido para el ataque de escaneo de puertos UDP.

	Árbol completo	Árbol reducido
Duración		
Puerto origen		✔
Puerto destino	✔	✔
Protocolo	✔	
Flag U		
Flag A		
Flag P		
Flag R		
Flag S		
Flag F		
Estado de reenvío		
Tipo de servicio		✔
Paquetes intercambiados	✔	
Número de bytes	✔	✔

Tabla 4.2 Parámetros utilizados en la construcción del árbol para el ataque escaneo UDP

El árbol completo va a depender del protocolo, ya que en este tipo de ataque solo se tiene en cuenta los puertos con protocolo UDP. El puerto destino viene delimitado por un rango desde puertos menores al 6059 hasta el 11835. Esto se debe a que en los instantes de tiempo en los que se han capturado los flujos de la red se estaba realizando un escaneo solo entre esos números de puertos. El escaneo UDP solo nos permite escanear aproximadamente un puerto por segundo. Los paquetes intercambiados tendrán que ser por lo general uno ya que son los que envía el atacante a cada puerto para determinar su estado actual.

En este caso, he querido añadir la representación del árbol reducido, ya que aporta más información y puede ser más útil que el árbol completo. El rango de escaneo se encuentra entre los puertos 6008 y 6076. Se considerará un ataque de escaneo de puertos cuando los flujos provengan de unos determinados puertos de origen, los cuales pertenecerán a la máquina atacante. Los atributos seleccionados son muy manipulables por lo que, tras analizar el árbol, nos quedariamos con el árbol completo.

- Scan11:

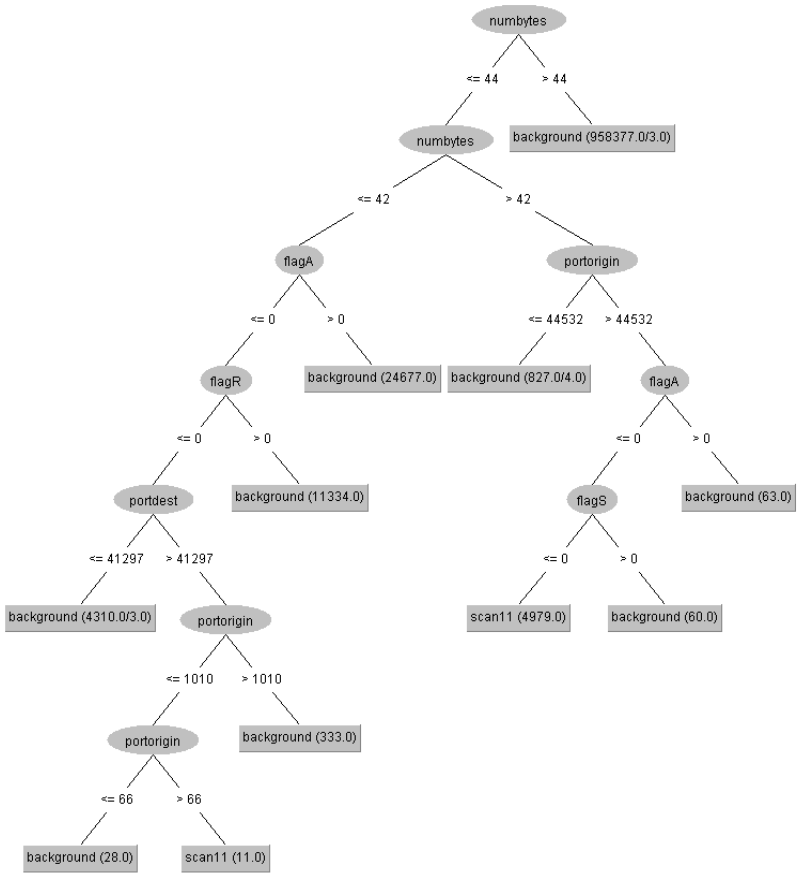


Ilustración 4.8 Árbol completo para el ataque de scan11.

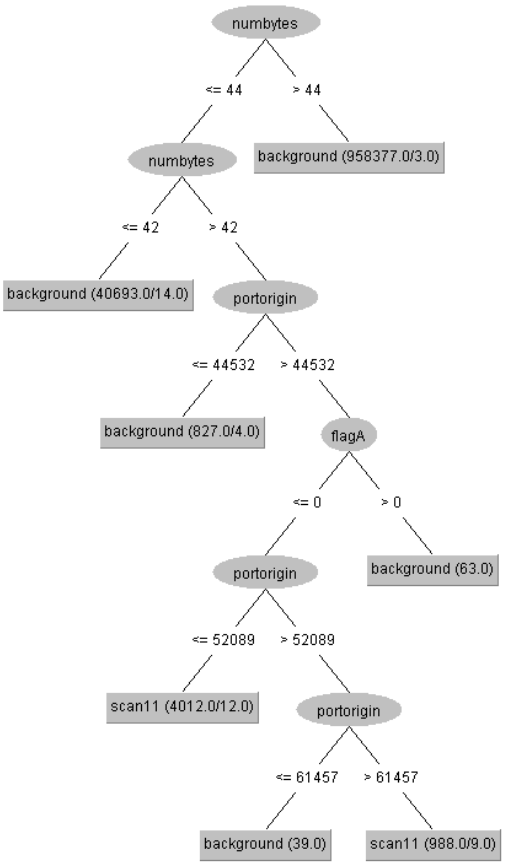


Ilustración 4.9 Árbol reducido para el ataque de scan11.

	Árbol completo	Árbol reducido
Duración		
Puerto origen	✔	✔
Puerto destino	✔	
Protocolo		
Flag U		
Flag A	✔	✔
Flag P		
Flag R	✔	
Flag S	✔	
Flag F		
Estado de reenvío		
Tipo de servicio		
Paquetes intercambiados		
Número de bytes	✔	✔

Tabla 4.3 Parámetros utilizados en la construcción del árbol para el ataque scan11

En este caso primero se decide en función del número de bytes. Sabemos que el tamaño mínimo de la cabecera UDP es de 8 bytes, por lo que el tamaño adicional será en función de los datos que transporte. En el caso de los Flag ACK y SYN, si estuvieran activados añadirian tamaño al número de bytes, ya que tienen que aparecer en la cabecera, es por ello que hay un rango tan específico del tamaño de los flujos. Filtrará en función de los puertos de origen y de destino.

En el árbol reducido, tenemos en cuenta el flag ACK que si estuviera activo haría que el número de bytes fuera mayor de 42 y clasificaría el flujo como tráfico normal. En función del puerto de origen acota nuestro rango de decisión obteniendo cuales son los puertos que utiliza la máquina atacante.

- Scan44:

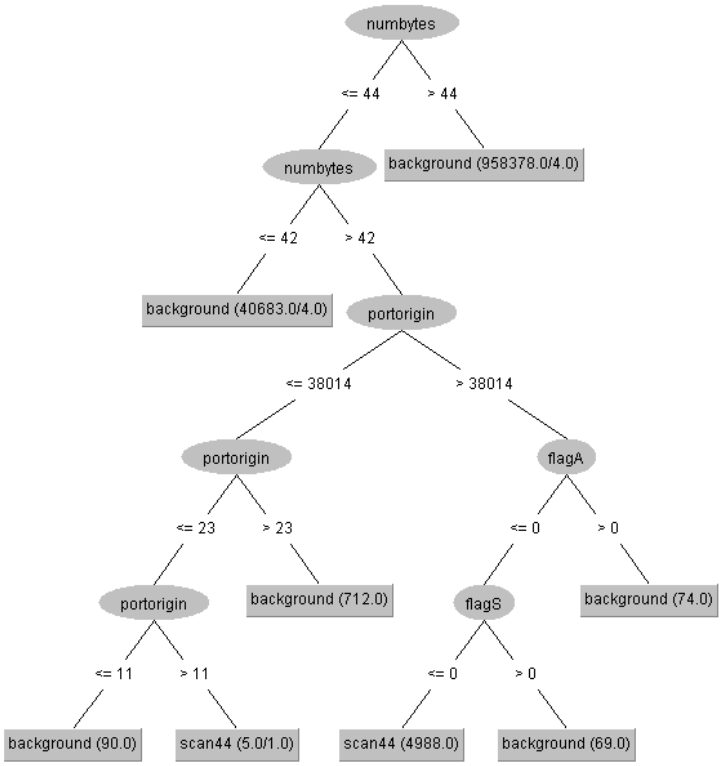


Ilustración 4.10 Árbol completo para el ataque de scan44.

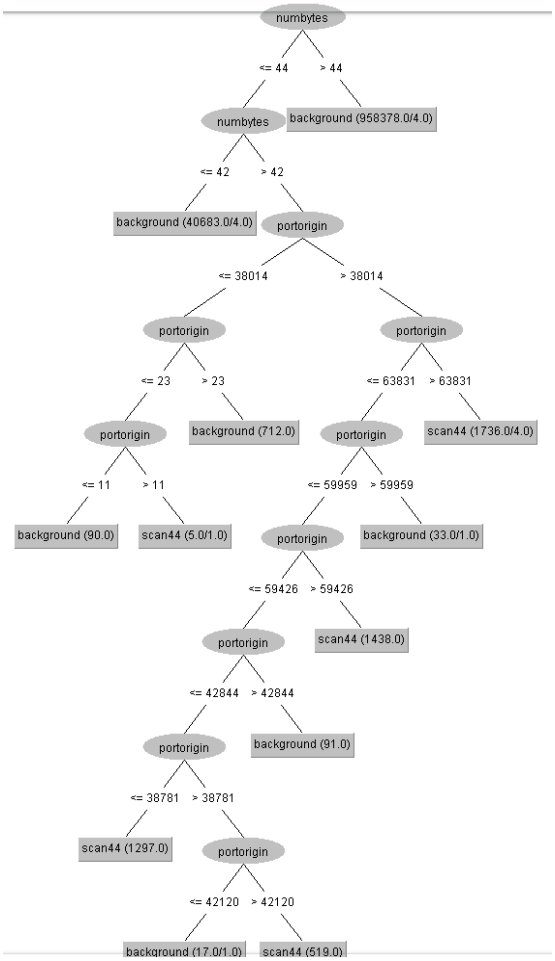


Ilustración 4.11 Árbol reducido para el ataque de scan44.

	Árbol completo	Árbol reducido
Duración		
Puerto origen	✔	✔
Puerto destino		
Protocolo		
Flag U		
Flag A	✔	
Flag P		
Flag R		
Flag S	✔	
Flag F		
Estado de reenvío		
Tipo de servicio		
Paquetes intercambiados		
Número de bytes	✔	✔

Tabla 4.4 Parámetros utilizados en la construcción del árbol para el ataque scan44

Este es un ataque muy similar al de scan11 por lo que el árbol obtenido es bastante similar. En este caso al realizarse el ataque desde varias maquinas hacia varias maquinas no tendrá en cuenta los puertos de destino. La mayoría de flujos se clasifican en función del número de bytes como podemos observar.

En el caso del árbol reducido ocurre lo mismo, se clasifican en función del número de bytes de cada flujo pero añade varios caminos para identificar con más precisión los puertos origen que utilizan las máquinas atacantes.

- Nerisbotnet:

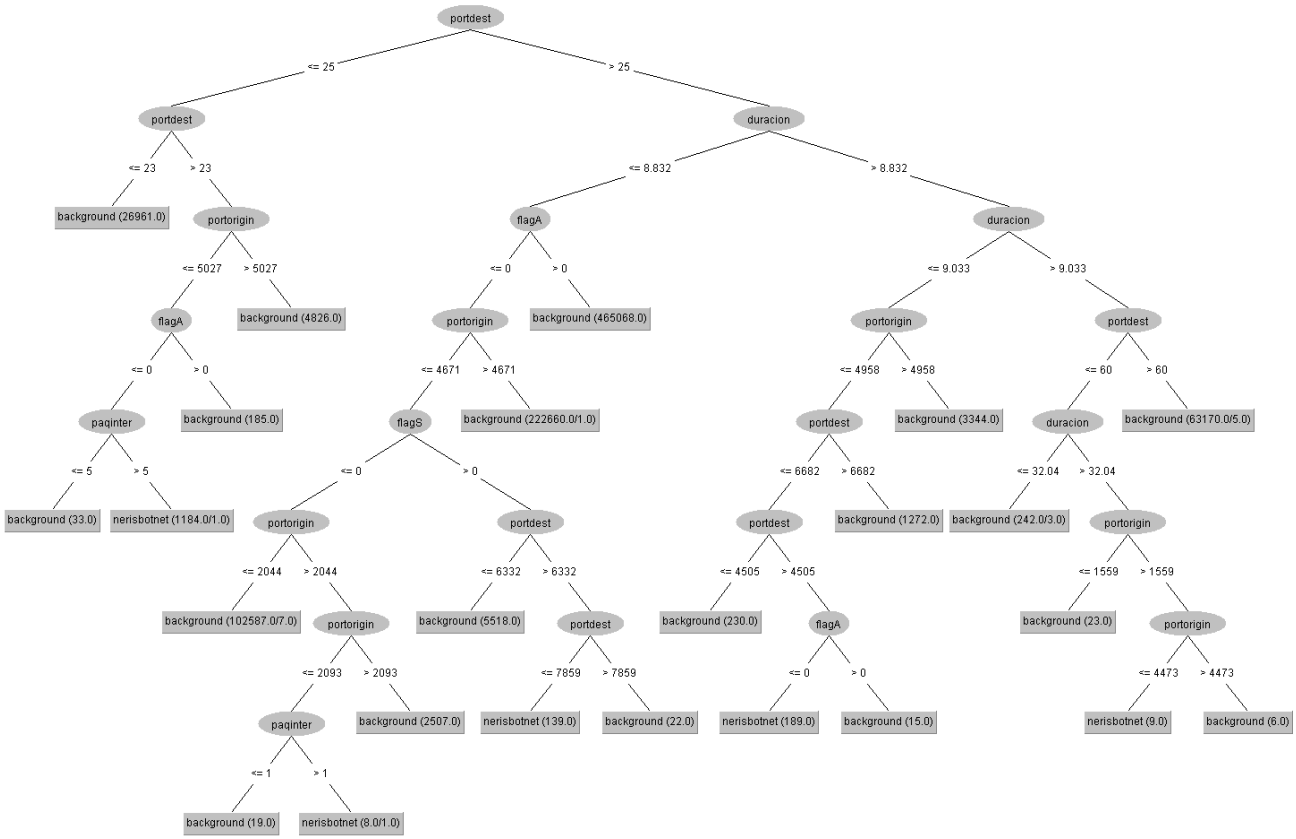


Ilustración 4.12 Árbol completo para el ataque de nerisbotnet.

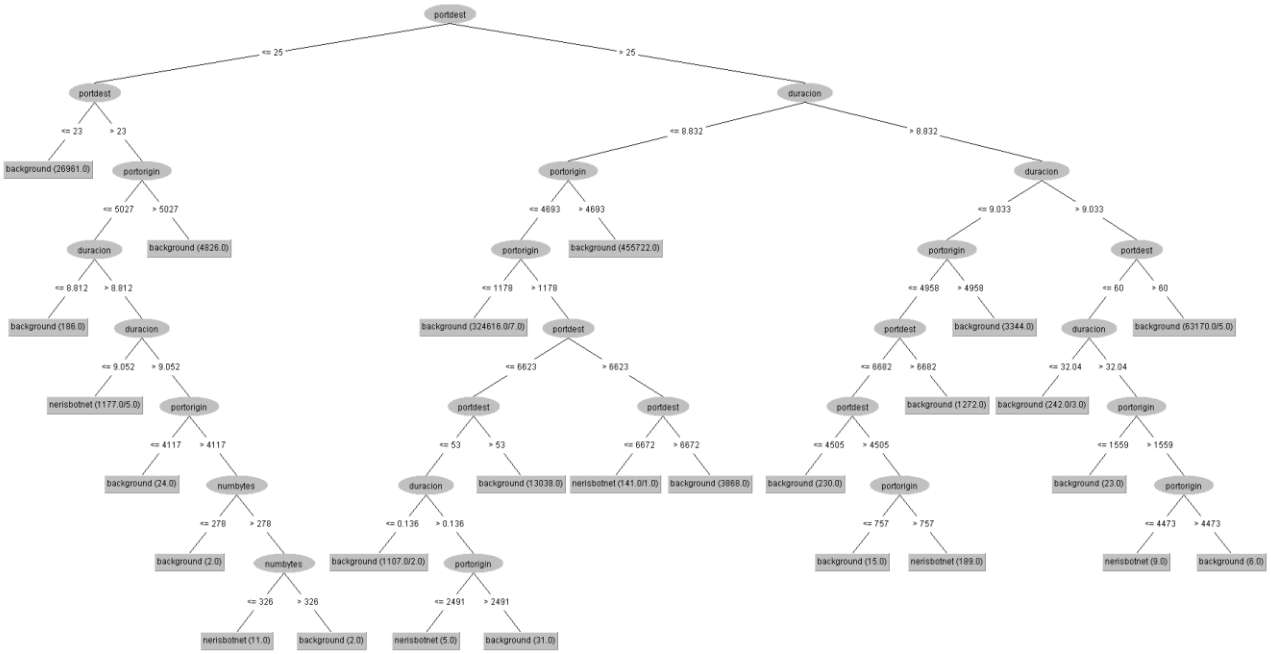


Ilustración 4.13 Árbol reducido para el ataque de nerisbotnet.

	Árbol completo	Árbol reducido
Duración	✔	✔
Puerto origen	✔	✔
Puerto destino	✔	✔
Protocolo		
Flag U		
Flag A	✔	
Flag P		
Flag R		
Flag S	✔	
Flag F		
Estado de reenvío		
Tipo de servicio		
Paquetes intercambiados	✔	
Número de bytes	✔	✔

Tabla 4.5 Parámetros utilizados en la construcción del árbol para el ataque nerisbotnet

Podemos comprobar que el árbol obtenido es mucho más grande que para los otros ataques, esto se debe a que el ataque de botnet es muy dificil de clasificar con certeza, ya que se desconoce el origen de las máquinas atacantes y el número de estas es también mucho más grande que en los demás casos.
La duración y los puertos de origen y destino variarán en función de cada máquina atacante. Pueden observarse máquinas que realizan distintas técnicas para llevar a cabo el ataque, por ello variarán la duración, el tamaño de los flujos y los paquetes intercambiados.

El árbol reducido es prácticamente igual, solo que no utiliza los flags ni el número de paquetes intercambiados ya que son atributos de menor peso dentro de este tipo de ataque. Será dificil determinar por qué en algunos flujos dichos flags aparecen activados, ya que las distintas técnicas usadas por las máquinas ocasionarán una gran variabilidad dentro de los datos.

4.5 Conclusiones

En este capítulo se han expuesto las diferentes pruebas y se han razonado los resultados obtenidos. A continuación, se muestra una tabla que recoge el porcentaje de aciertos en función del tipo de ataque y del tipo de árbol escogido, junto a su tamaño resultante y al tiempo empleado en construir dichos árboles:

	Porcentaje de aciertos del árbol completo	Porcentaje de aciertos del árbol reducido	Tamaño total del árbol completo	Tamaño total del árbol reducido	Tiempo tomado en crear el árbol completo	Tiempo tomado en crear el árbol reducido
DoS	99.9998 %	99.9965 %	18	9	15.49	3.22
Scan UDP	100 %	100 %	18	9	15.94	5.26
Scan11	99.999 %	99.9958 %	21	13	26.5	5.74
Scan44	99.9991 %	99.9958 %	15	23	29.75	3.52
Nerisbotnet	99.998 %	99.9974 %	47	51	38.38	11.14

Tabla 4.6 Valores obtenidos en función del tipo de ataque

A raíz de los resultados hemos comprobado que hay casos en los que el árbol reducido acaba siendo más grande que el normal, esto ocurrirá sobre todo en los ataques más difíciles de clasificar. Por lo general la cantidad de aciertos suele disminuir utilizando el árbol reducido, pero es una disminución que nos podemos permitir, ya que es casi inapreciable frente a la gran diferencia de tiempos que existe entre la creación de los dos árboles. Este será un punto importante de nuestro estudio, ya que es muy importante la rapidez con la que se detecten los ataques. Como podemos observar el tiempo que tarda en generar el árbol completo se reduce en un 80% aproximadamente para la mayoría de los casos, al utilizar los parámetros de mayor peso.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

En este apartado se van a exponer las conclusiones globales obtenidas a partir del estudio realizado.

Partiendo de unos datos generados por la Universidad de Granada, los cuales presentan tráfico normal mezclado con ataques generados sintéticamente, hemos conseguido analizar teóricamente cuáles podrían ser los parámetros que más información relevante nos aportasen para clasificar el tráfico. Gracias a los programas desarrollados y a las pruebas realizadas hemos podido maximizar el rendimiento del algoritmo y crear un conjunto de datos, en el cual menos del 1% del tráfico es de ataque, para así asimilarlo lo máximo posible a una situación real.

A partir de nuestra hipótesis inicial [2] hemos conseguido prácticamente un 100% de aciertos en todos los tipos de ataque, al igual que se expone en el artículo referenciado. Por ello podemos verificar que, tras haber evaluado otros algoritmos, el árbol J48 es muy eficaz a la hora de clasificar intrusiones detectadas en un conjunto de tráfico de red.

Para finalizar, cabe mencionar que los árboles han sido entrenados con conjuntos de datos extraídos de una misma captura realizada a lo largo de varios meses, por lo que es probable que su porcentaje de aciertos disminuya al utilizar datos externos a dicha captura. De todas maneras, se ha conseguido razonar de manera lógica la obtención de los parámetros utilizados en cuestión por lo que esto aportará fiabilidad a la hora de utilizar otros tipos de datos.

5.2 Trabajo futuro

De cara a continuar desarrollando este estudio en el futuro, se identifican distintas tareas para aumentar su alcance:

- Generar distintos árboles utilizando capturas de datos de diferentes procedencias y evaluarlos.

- Introducir otros tipos de ataques para así poder obtener sus parámetros más característicos.

- Conseguir generar un árbol que sea capaz de identificar varios ataques de red a la vez en función del tráfico de entrada.

Referencias

[1] https://www.revistatransformaciondigital.com/2017/10/09/el-75-de-los-ciberataques-no-son-detectados/ consultado el día 16 de junio de 2019

[2] Bashir, U., & Chachoo, M.A. (2017). Performance Evaluation of J48 and Bayes Algorithms for Intrusion Detection System. *International Journal of Network Security & Its Applications, 9*, 01-11.

[3] https://www.redeszone.net/2010/11/03/ataques-a-las-redes-listado-de-diferentes-ataques-a-las-redes-de-ordenadores/ consultado el día 7 de junio de 2019

[4] https://es.wikipedia.org/wiki/Conjunto_de_datos consultado el día 7 de junio de 2019

[5] Maciá-Fernández, G.; Camacho, J.; Magán-Carrión, R.; Fuentes-García, M.; García-Teodoro, P.; Theron, R. (2018). UGR'16: Un nuevo conjunto de datos para la evaluación de IDS de red. En XIII Jornadas de Ingeniería telemática (JITEL 2017). Libro de actas. Editorial Universitat Politècnica de València. 71-78. doi:10.4995/JITEL2017.2017.6520

[6] J. L. García-Dorado, J. E. López, J. Aracil, V. López, J. A. Hernández, S. López-Buedo y L. de Pedro. "Utilidad de los flujos NetFlow de RedIRIS para análisis de una red académica"

[7] https://es.wikipedia.org/wiki/%C3%81rbol_de_decisi%C3%B3n consultado el 8 de junio de 2019

[8] https://es.wikipedia.org/wiki/Aprendizaje_basado_en_%C3%A1rboles_de_decisi%C3%B3n consultado el 8 de junio de 2019

[9] http://www.locualo.net/programacion/mineria-datos-weka-ficheros-arff/00000019.aspx consultado el 8 de junio de 2019

[10] https://es.wikipedia.org/wiki/Weka_(aprendizaje_autom%C3%A1tico) consultado el 8 de junio de 2019

[11] https://es.wikipedia.org/wiki/C4.5 consultado el 8 de junio de 2019

Anexos

A Matrices de confusión

- Ataque Denegación de Servicio (DoS):

```
=== Confusion Matrix ===

      a      b   <-- classified as
 999997      2 |      a = background
      0   5000 |      b = dos
```

- Escaneo de puertos UDP:

```
=== Confusion Matrix ===

      a      b   <-- classified as
1000000      0 |      a = background
      0   5000 |      b = anomaly-udpscan
```

- Scan11:

```
=== Confusion Matrix ===

      a      b   <-- classified as
 999999      0 |      a = background
     10   4990 |      b = scan11
```

- Scan44:

```
=== Confusion Matrix ===

      a      b   <-- classified as
 999998      1 |      a = background
      8   4992 |      b = scan44
```

- Botnet:

```
=== Confusion Matrix ===

      a      b   <-- classified as
 898672      2 |      a = background
     16   1527 |      b = nerisbotnet
```